

# Programozási alapismeretek (C)

Dobos László

Komplex Rendszerek Fizikája Tanszék

2018. szeptember 10.

Dobos László

- ▶ Komplex Rendszerek Fizikája Tanszék
- ▶ É. 5.60
- ▶ `dobos@complex.elte.hu`

A tantárgy weboldala:

- ▶ <http://www.vo.elte.hu/~dobos/teaching/progalap2018>
- ▶ facebook csoport
- ▶ youtube csatorna

# Órák időpontjai

Előadás: hétfő 17-19 – É 0.79

- ▶ összesen 7 előadás!
- ▶ utolsó előadás: nov 5.

Gyakorlatok – É 5.56

- ▶ kedd 12:15 - 13:00
- ▶ kedd 16:15 - 17:00
- ▶ csütörtök 9:15 - 10:00
- ▶ csütörtök 11:15 - 12:00

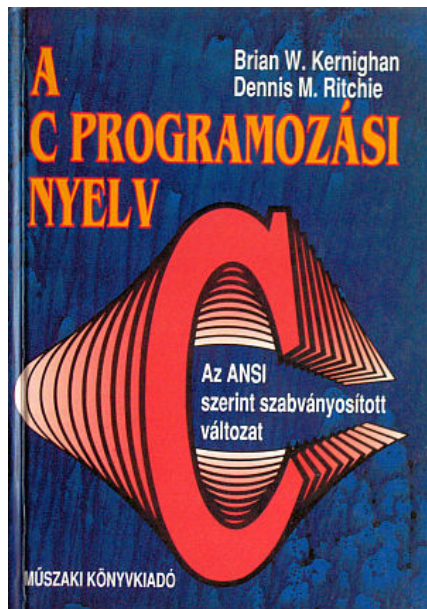
A gyakorlatok látogatása kötelező

- ▶ katalógus lesz
- ▶ az első héten is lesznek gyakorlatok

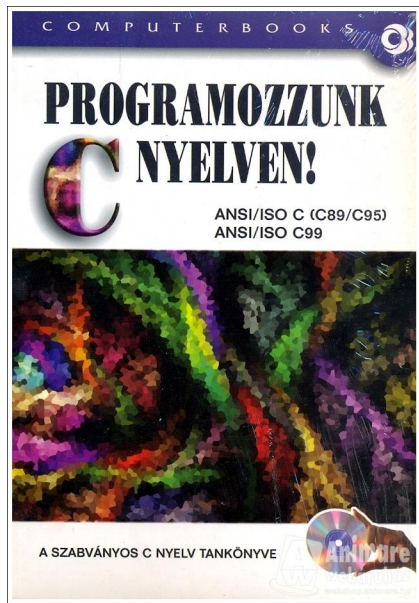
Biztosan nem lesz óra:

- ▶ okt. 22-23., okt. 29 - nov. 2.

Ajánlott könyv: K&R: A C programozási nyelv



Ajánlott könyv: Programozzunk C nyelven



1. Rövid történeti bevezető
2. A C nyelv utasításai
3. Függvények, strukturált programozás
4. Változók és típusok, struktúrák
5. Operátorok, kifejezések
6. Memóriakezelés, verem, halom, mutatók
7. Egyszerű adatmodellek: vektor, mátrix, láncolt lista
8. Fájlkezelés, szöveges fájlok, formátumozás
9. Egyszerű algoritmusok: keresés, sorba rendezés
10. Stringek C-ben
11. Lineáris egyenletrendszerek megoldása Gauss-eliminációval
12. Közönséges differenciálegyenlet-rendszerek numerikus megoldása

# Követelmények

A félév során 6 kötelezően beadandó házi feladat

- ▶ ezeken otthon előre kell dolgozni
- ▶ a gyakorlatokon is ezeket oldjuk meg
- ▶ segítünk a megoldásban és a hibakeresésben

Gyakorlatokra készülés

- ▶ mindig legyen ott az aktuális feladathoz tartozó félkész program
- ▶ google drive-ra, caesarra feltölteni! USB nincs!
- ▶ saját laptopot lehet használni

Tetszőleges C fordító használható C99 szabványig bezárólag

- ▶ nem C++
- ▶ Ajánlott környezet: Code::Blocks
- ▶ Visual Studio, VS Code, XCode, CLion, Eclipse stb.
- ▶ a programnak végül `gcc -std=c99 *.c` parancssorral fordulnia kell!

## A házi feladatok értékelése

A feladatok két részből állnak:

- ▶ kötelező **alapfeladat**, ami feltétlen beadandó a ketteshöz
- ▶ **teljes feladat**: jobb jegy szerzéséhez minél többet meg kell oldani

Hirtelen halál:

- ▶ ha valakit bizonyíthatóan másoláson érünk, nem kap jegyet
- ▶ szóban konzultálni lehet, programot megosztani egymással nem

Beadási határidők:

- ▶ minden feladatra van egy elsődleges határidő
- ▶ késés esetén fekete pont
- ▶ hibás megoldás javítását később is be lehet küldeni
- ▶ teljes megoldást a félév során bármikor be lehet küldeni

Legvégső beküldési határidő: **2018. december 19. éjfélig**



# Házi feladatok

## 1. Határidő: 2018. szeptember 23. 23:59

Készítsünk programot, ami kilistázza a püthagoraszai számhármassokat!

- a) Egy megadott  $n$  számig úgy, hogy  $a, b, c \leq n$  (A)
- b) Az első  $n$  hármast úgy, hogy a számhármassok ismétlődése megengedett. (A)
- c) Az első  $n$  hármast úgy, hogy a számhármassok ismétlődése nem megengedett. A számhármassokat akkor tekintjük ismétlődőnek, ha a számok sorba rendezve megegyeznek, pl.  $(3\ 4\ 5) = (4\ 5\ 3)$ . (T)
- d) Az első  $n$  hármast úgy, hogy a számhármassok nem lehetnek egymás számszorosai, azaz  $(3\ 4\ 5) = (6\ 10\ 8)$  nem megengedett. (T)

A program paraméterét parancssori argumentumként adjuk meg: ha az első paraméter 'a', az a) feladatrészt, ha 'b' a b) feladatrészt stb. megoldását adja vissza; a második paraméter legyen  $n$  értéke.

## 2. Határidő: 2018. október 7. 23:59

Készítsük programot, amely egy fájlból beolvassa egy vektor elemeit! A fájl neve és a vektor mérete legyen parancssori paraméter! A program ezek után

- a) írja ki a vektor legnagyobb elemének indexét és értékét! (A)
- b) írja ki a vektor minden második elemét indexek szerint visszafelé haladó sorrendben! (A)
- c) rendezze sorba a vektor elemeit quick sort segítségével az stdlib qsort függvényével, illetve saját quick sort implementációval (T)
- d) minél kevesebb lépés elvégzésével írja ki, hogy egy parancssori paraméterként megadott szám szerepel-e a vektor elemei között, és ha igen, hányadik indexnél (T)

## Házi feladatok

### 3. Határidő: 2018. október 21. 23:59

Készítsünk programot, amely két szöveges, szóközökkel és újsor-karakterekkel tagolt fájlból beolvas egy-egy mátrixot, és kiszámítja a mátrixok szorzatát! A program ellenőrizze a mátrixok összeszorozhatóságát is!

- a) A mátrixokat tartalmazó fájlok neve, illetve a mátrixok mérete legyen parancssori paraméter! (A)
- b) Írjuk meg a programot úgy, hogy ne kelljen megadni a mátrixok méretét, hanem azt a program automatikusan találja ki! Működjön a beolvasó rutin minden olyan fájlra, amit szóközök és újsor-jelek tagolnak, és egy szövegszerkesztőben megnyitva mátrixként értelmezhetőek! (T)

### 4. Határidő: 2018. november 11. 23:59

Készítsünk programot, mely megvalósítja a Gauss-Jordan-elimináció algoritmusát

- a) részleges pivotálással (A).
- b) Az elimináció során a program detektálja, ha a mátrix szinguláris! (T)
- c) Írjuk meg az algoritmust részleges pivotálás helyett teljes pivotálással is! (T)
- d) Oldjuk meg a mátrixinvertálás problémáját saját C függvény megírása helyett a LAPACK csomag megfelelő függvényével. Hasonlítsuk össze a saját megoldás és a LAPACK sebességét, illetve skálázását. (T)

## Házi feladatok

### 5. Határidő: 2018. november 23. 23:59

Készítsük programot, mely egy fájlból előre ismeretlen számú valós számot olvas be, és azt egy láncolt listában tárolja. Miután a számok már a memóriában vannak

- a) Egy min és max értéket parancssori paraméterként megadva a program írja ki azokat a számokat, melyek min és max közé esnek! (A)
- b) Határozzuk meg, hogy egy parancssori paraméterként beadott szám hányadik eleme az eredeti vektornak! (A)
- c) Rendezzük sorba a számokat a tört részüik szerint növekvő sorrendben! (T)
- d) Oldjuk meg a feladatot láncolt lista helyett keresőfa segítségével! (T)

### 6. Határidő: 2018. december 9. 23:59

Készítsünk programot, amely alkalmas közös differenciálegyenlet-rendszerek integrálására, és a Hold Föld körüli mozgásának szimulációjára. A megvalósítás során célszerű a változókat egységesen, egyetlen tömbként kezelni, hiszen így a program tetszőleges egyenletrendszer integrálására alkalmas lesz. Írjuk meg úgy az integrátort, hogy a megoldandó egyenletrendszert függvényparaméterként várja!

- a) Valósítsuk meg a feladatot az Euler-módszer segítségével (A)
- b) Ábrázoljuk a Hold Föld körüli pályáját és a rendszer teljes energiát az idő függvényében! (A)
- c) Hasonlítsuk össze az Euler-módszert a 4-rendű adaptív Runge-Kutta-módszerrel. (T)
- d) Integráljuk a Lorenz-féle egyenletrendszert és ábrázoljuk az eredményeket több vetületben. (T)